

# Advanced Object-Oriented Features

By André Rodrigues (xrodri10) and William Costa (xcosta07)

Object-oriented programming languages bring many useful features for programmers that conventional non-object-oriented languages don't have. These features, such as constructors, polymorphism, inheritance, etc allow programmers to write readable and well-organized code, easily extensible.

The main focus of this presentation will be constructors and their advanced features and generics and how they can be useful as a tool in object-oriented programming.

The function of a constructor is mostly to initialize the memory allocated when the *new* keyword is encountered. An advanced technique which emerged from this concept is multiple constructors, that can be used when the programmer wants to offer multiple ways to define an object. This can be useful in a library, for example, when we want the option to define some starting values for a certain object instead of using the default ones.

Another powerful advanced feature is generics. The aim of generics is to give programmers the possibility of writing type independent code. With this, we can create versatile methods, for instance, a method that can sort an array of integers/strings/an object of any sortable class. This allows for more extensibility and clearer code, which will prove to be very important in the long run, when the code base is significantly larger.